



# MERN STACK

Career Path

Blueprint

হাসিন হায়দার

CTO, Happy Monster  
Founder, Learn With Hasin Hayder

সুমিত মাহা

Founder & CEO at Learn with Sumit  
Co-Founder  
& Managing Director at Analyzen

আসিফ মাহির

Software Developer,  
JS Bangladesh

সাইদুর রহমান সেতু

Founder  
JS Bangladesh

সীমাত্ত পাল

Software Engineer,  
Markopollo.ai &  
NextGen Cloud

ইফতিখার রহমান

Software Engineer,  
ACI Ltd.

# Overview

ওয়েব ডেভেলপমেন্টের ইন্ডাস্ট্রি প্রতিনিয়তই নতুন নতুন টেকনোলজি আসছে। আসছে নতুন নতুন স্ট্যাক। MEAN, MERN, LAMP, PERN, MEVN-সহ আরও অনেক স্ট্যাক। **M-MongoDB, E-ExpressJS, R-ReactJS, N-NodeJS** মিলেই হচ্ছে MERN স্ট্যাক। একটি মাত্র ল্যাঞ্চেজ শিখে কম্পিউট করে ফেলা যায় সম্পূর্ণ স্ট্যাক। বর্তমানে আমাদের দেশে মার্ট স্ট্যাক ওয়েব ডেভেলপারদের চাহিদা বেড়েই চলছে। ইন্ডাস্ট্রি এক্সপার্টদের নলেজ এবং বিগত সফল ২ ব্যাচের এক্সপেরিয়েন্সের উপর বেইজ করে ডিজাইন করা দেশসেরা আউটলাইনে আপনি পাবেন একজন ইন্ডাস্ট্রি রেডি ডেভেলপার হওয়ার কম্পিউট গাইডলাইন।

আমাদের এই ক্যারিয়ার পাথে ইন্ট্রোকটর হিসেবে থাকছেন Learn With Hasin Hayder এর ফাউন্ডার দ্বা ওয়ান এন্ড অনলি **হাসিন হায়দার**, Learn With Sumit এর ফাউন্ডার **সুমিত সাহা**, Markopollo.ai আর NextGen Cloud-এর সফটওয়্যার ইঞ্জিনিয়ার **সীমান্ত পাল**, JS Bangladesh-এর ফাউন্ডার **সাইদুর রহমান সেতু**, তাদের সাথে থাকছেন একই কোম্পানির সফটওয়্যার ডেভেলপার আসিফ মাহির, ACI Limited এর সফটওয়্যার ইঞ্জিনিয়ার **ইফতিখার রহমান**।

# ক্যারিয়ার পাথে কী কী শিখবেন?

- ঔয়েব ডেভেলপমেন্ট বেসিকস
- HTML, CSS, Bootstrap
- জাভাস্ক্রিপ্ট ও এর ফ্রেমওয়ার্ক
- React JS
- Express JS
- Node JS
- MongoDB

৩০০+  
প্রিৱেকেডেড ভিডিও

৭০+  
লাইভক্লাস

জেইলি ৩টি  
সামোর্ট সেশন

১০০+ কোম্পানিতে  
চাকরির সুযোগ

১২+ প্রজেক্টস  
ও অ্যাসাইনমেন্ট

উৎকলি  
প্রজেক্ট ডে

মক  
ইন্টারভিউ

লাইফটাইম  
অ্যাকসেস

 দেশসেরা জব প্ল্যাটফর্মেন্ট টিমের সামোর্ট ১০০+ কোম্পানিতে জব  
ও ইন্টার্ভিউসের সুযোগ

# ক্যারিয়ার পাথটি কাদের জন্য?

ওয়েব  
ডেভেলপমেন্টে  
আগ্রহী

ইউনিভার্সিটি  
স্টুডেন্ট

ফ্রিল্যান্সার

ফ্রেন্টেন্ড ও ব্যাকেন্ড  
ডেভেলপমেন্ট  
নিয়ে কাজ করতে চান

মেইন কথা হচ্ছে, MERN স্ট্যাক ওয়েব ডেভেলপমেন্ট নিয়ে আপনার ইন্টারেস্ট থাকলে,  
সেই ইন্টারেস্টকে কাজ লাগিয়ে ক্যারিয়ার বিল্ড আপ করার দায়িত্ব আমাদের।

ক্যারিয়ার পাথে এন্রোলমেন্টের জন্যে আগে থেকে কি কিছু জানা লাগবে?

মোটেই না! আমাদের ক্যারিয়ার পাথ ডিজাইন করা হচ্ছে এমনভাবে  
যেন আপনার জানিং শুরু হয় একদম বেসিক থেকে। ওয়েব পেইজ কী  
থেকে জানিং শুরু হয়ে বিভিন্ন লার্নিং স্টেশন মার হয়ে আমরা পৌছাবো  
জব ল্যান্ডিংয়ে।



ক্যারিয়ার পাথে কোন কোন ল্যান্ডিং পাইজ শেখানো হবে?

- ওয়েব ডেভেলপমেন্টের জন্য বেসিক দুটি ল্যান্ডিং পাইজ তা থাকবেই: HTML-CSS. এর মধ্যে আমরা বুটস্ট্র্যাম শিখবো।
- ফন্টএণ্ডের জন্য জাভাস্ক্রিপ্ট আর এর ফ্রেমওয়ার্ক ReactJS
- ব্যক্টরের জন্য পাইথন আর এর ফ্রেমওয়ার্ক Django
- ডেটাবেস ম্যানেজমেন্টের জন্য mySQL

মোট কথা, একজন ইন্ডাস্ট্রিয়েলি ফুলস্ট্যাক ডেভেলপার হওয়ার জন্য সবধরনের স্কিলেই আপনাকে দক্ষ করে তোলা হবে। এই দায়িত্ব আমাদের; আপনার কাজ শুধু লম্বা এই জারিতে আমাদের সাথে থাকা।



কী কী প্রজেক্টে করবো?



#### • Frontend Projects:

##### • HTML-CSS:

- Personal Portfolio
- Blog Website
- Responsive Movie Landing Page
- E-Commerce Landing Page

##### • React:

- Basic React JS App
- Create a Website using React JS Hooks and Fetch API
- Create React JS App using Firebase Authentication
- Full Stack MERN Project using Payment Gateway Integration

##### • JavaScript:

- Functional Countdown Project
- Functional E-Commerce Shopping Cart
- Build 3 Websites using APIBook List



#### • Node JS Project

- Full Stack App Backend API

# ক্যারিয়ার পাথ নিয়ে আপনাদের সকল জিজ্ঞাসা

## Early Bird রেজিস্ট্রেশন জিজ্ঞাসা



EarlyBird রেজিস্ট্রেশন ক্যাম্পেইন অফার কতদিন থাকবে?

| এই ক্যাম্পেইন অফারটি (৯-২৪ জুন) থাকবে।



Early Bird এনরোলমেন্টে লার্নাররা কি ধরণের অফার পাবে?

| রেজিস্ট্রেশন করে আর আপনাকে অমেন্ট্র করতে হবেনা ফ্লাস  
শুরুর জন্য। এখন Early bird রেজিস্ট্রেশনকারি প্রত্যেকের জন্য  
থাকছে -

- ৩০০+ প্রিয়েকর্ডেড ভিডিও
- Early সামোর্ট সেশন
- লাইভ ফ্লাস
- ক্যারিয়ার গাইডলাইন
- প্রবলেম সলভিং
- নেটওয়ার্কিং
- ক্যারিয়ার বুস্ট আপ টিপস
- ফ্রি ডিনার
- MERN টি শার্ট



নেটওয়ার্কিং সেশন এবং মেট'রদের সাথে ফ্রি ডিনারের অপরচুনিটি  
কয় জন পাবে?

| সৌভাগ্যবান ১৫ জন এই অপরচুনিটি পাবে।



Early সামোর্ট সেশন কি প্রতি সপ্তাহে থাকবে?

| প্রতি সপ্তাহে ১ দিন।



এক্সটেন্সিভ কেয়ারে কয়টি লাইভ ফ্লাস হবে?

| এক্সটেন্সিভ কেয়ারে প্রতি ২ সপ্তাহে একটি করে লাইভ ফ্লাস থাকবে।



এক্সটেন্সিভ কেয়ারে লাইভ ফ্লাস কে নিবেন?

| আমাদের এক্সপার্ট মেন্টর ম্যানেল।

## ক্যারিয়ার পাথ জিঞ্জামা



কোন ইন্ট্রোকটর কোন টপিক পড়াবে?

| Javascript: সেতু ভাই, হাসিন ভাই।

| React: সুমিত সাহা ভাই, আসিফ মাহির ভাই।

| Node.Js: সীমান্ত পাল ভাই এবং সুমিত ভাই।

| Database: হাসিন ভাই এবং ইফতিখার ভাই।



ক্লাস কবে শুরু?

| ৬ আগস্ট, ২০২৪।



কোর্সটি কি বিগিনার ফ্রেন্ডলি?

| জুঁ, কোর্সটি বিগিনার ফ্রেন্ডলি। আপনার বেসিক স্ট্রাং করার জন্য, ক্যারিয়ার পাথ শুরুর আগে Early bird রেজিস্ট্রেশন করে প্রিমিয়াম ভিডিওগুলো দেখা শুরু করতে পারেন। এখানে থাকছে Early সামোর্ট সেশন এবং লাইভ ক্লাস।



ক্যারিয়ার পাথের কোনো ফ্রি ডেমো ভিডিও আছে?

| হঁয়, সুমিত ভাই এবং হাসিন ভাইয়ের ফ্রি ডেমো ভিডিও দেখতে এখানে ক্লিক করুন।



MERN ক্যারিয়ার পাথ ব্যাচ ৩- এ কী কী বাব লাইভ ক্লাস হবে?

| মঙ্গলবার, বৃহস্পতিবার, শুক্রবার - রাত ৯ টায়।



উইকলি লাইভ ক্লাসগুলো কে নিবেন?

| লাইভ ক্লাসগুলো হাসিন ভাই, সুমিত ভাই, সেতু ভাই, মাহির ভাই এবং সীমান্ত ভাই নিবেন।

### সামোর্ট সেশন কখন এবং কবে হবে?

প্রতি সপ্তাহে **৬ দিন, ৩ বেলা। দুপুর, সন্ধ্যা এবং রাতে।** তাই আপনি যখনই ফ্রি থাকবেন নিজের প্রবলেমগুলোর ইন্ট্যাক্ট সলিউশন মেঘে যাবেন আমাদের এক্সপার্ট সামোর্ট ইন্ট্রোকটরদের কাছে।

### প্রজেক্ট ডে কি?

পুরো ক্যারিয়ার মাথে ইন টোটাল **২৪ টি প্রজেক্ট ডে** থাকবে। প্রজেক্ট রিলেটেড সব প্রবলেম এই প্রজেক্ট-তে সল্ভ করা হবে। আমরা একটি আউটপুট ফোকাসড লার্নিং সিস্টেমে বিশ্বাস করি তাই প্রজেক্টে রিলেটেড যেকোনো সমস্যার সমাধানের উদ্দেশ্যেই একটি ফুল প্রজেক্ট ডে-তে ডেভিলপেটেড সামোর্ট ইন্ট্রোকটররা আপনাকে হেল্প করতে প্রস্তুত থাকবেন।

### শুধু কি ব্যাক-এন্ড ফোকাস করা হবে?

না। ফ্রন্ট এন্ড এবং ব্যাক-এন্ড দুইদিকে সমান ফোকাস করা হবে।

### কোন ধরনের ক্যারিয়ার সেশন/ ইন্টারভিউ প্রিমারেশন থাকবে?

প্রতিটি ইস্পরট্যান্ট টপিক শেষে আপনার নলেজ খালিয়ে নিতে থাকছে মক ইন্টারভিউ, এর সাথে জব মডিউলের সময় মেঘে যাচ্ছেন একটি **ইন্টারভিউ হ্যান্ডবুক সম্পূর্ণ ফ্রি।** এছাড়াও ক্যারিয়ার সেশন থাকছেই। তাই ক্যারিয়ার মাথে ভর্তি হওয়ার পর আপনাকে জব রেডি করে তোলার জন্য যা যা প্রয়োজন সবকিছু নিয়ে আমরা রেডি।



### ক্যারিয়ার পাথ শেষে জব প্ল্যাসমেন্টের ব্যবস্থা আছে?

হ্যাঁ, আমাদের এই লম্বা ক্যারিয়ার পাথের লাস্ট মডিউলটিই হচ্ছে জব প্রিপারেশনের। তিনি সম্ভাব্যে এই মডিউলে আপনাকে CV আৰ পোর্টফোলিও বানানো থেকে শুরু করে বিভিন্ন টেকনিক্যাল ইন্টারভিউ হ্যান্ডেল করা পর্যন্ত সবকিছু আপনাকে শেখানো হবে এবং অন্তে মিভালি। পাশাপাশি আমাদের আফিলিয়েটেড ১০০+ কোম্পানিতে পাচ্ছেন **জব+ইন্টার্ভিউ প্ল্যাসমেন্ট** অপরচুনিটি।



### ক্যারিয়ার পাথে এন্টারেল করতে কত টাকা লাগবে?

১০০০ টাকা দিয়ে এন্টারেল করে পেয়ে যান ক্যারিয়ার বিল্ড আপের জন্যে দেশসেৱা গাইডলাইন।



### ক্যারিয়ার পাথটি কীভাবে কিনবো?

ওয়েবসাইটে অ্যাড টু কার্ট থেকে কিনতে পারবেন, তাৰাড়াও রেজিস্ট্রেশন ফর্মের মাধ্যমে ম্যানুয়ালিও কিনতে পারবে।

ওয়েবসাইট লিংকঃ **এখানে ক্লিক করুন**  
গুগল রেজিস্ট্রেশন ফর্মঃ **এখানে ক্লিক করুন**



### ক্যারিয়ার পাথ কোন ডিসকাউন্ট কি থাকছে?

জু থাকছে, ডিসকাউন্টের জন্য আমাদের **ফেসবুক পেইজে** মেসেজ করুন অথবা এই নম্বৰে কল করুন: **09613-829043**



### কীভাবে পেমেন্ট করবো?

বিকাশে **01727659043/ 01763881476** -এই ২টি নম্বরের সেন্ড মানি অপশনের মাধ্যমে পেমেন্ট করতে পারবে। আর নগদে **01727659043** এই নম্বরের সেন্ড মানি অপশনের মাধ্যমে পেমেন্ট করতে পারবে। ওয়েবসাইট থেকে কিনলে ভিসা কার্ড, মাস্টার কার্ড, বিকাশ, নগদ যে কোন মাধ্যমেই পেমেন্ট করা যাবে। এছাড়াও আপনি দেশের বাইরে থেকে পেমেন্ট করতে চাইলে Stripe -এ পেমেন্ট করতে পারবেন।



### ক্যারিয়ার পাথের উত্তি বাতিল করা কি সম্ভব?

না, ক্যারিয়ার পাথ এনরোলমেন্টের পরে তা আর বাতিল করার অপশন নেই।



### কোন টেকনিকাল সমস্যা কীভাবে রিপোর্ট করবো?

ওয়েবসাইটের হেল্প অপশনে টেক্সট দিতে পারেন কিংবা আমাদের ফেইসবুক মেইজে আমাদের রিচ করতে পারেন। এছাড়াও যেকোনো প্রয়োজনে **09613-829043** এই নম্বারে কল করুন।



# Detailed Course Outline



# MERN Course Syllabus

## HTML & CSS

**Upon successful completion of these 4 weeks, students will be able to:**

- Understand the fundamental concepts of web development and the roles of HTML and CSS.
- Write clean, semantic HTML code to structure web page content.
- Style web pages using CSS, incorporating basic layout techniques, typography, and colors.
- Implement responsive design principles to ensure websites adapt to different screen sizes.
- Understand and utilize CSS frameworks like Bootstrap and Tailwind CSS for rapid website development.
- Explore and understand the basics of CSS preprocessors and alternative CSS frameworks.
- Utilize CSS Grid and Flexbox for creating complex and responsive layouts.
- Be prepared for further learning and exploration in web development.

# HTML & CSS

## Week 01: HTML Foundations

- **Class - 01: Introduction to the Web & HTML**
  - What is web development?
  - Front-end vs. back-end development
  - Introduction to HTML: tags, elements, and attributes
  - Basic HTML structure: <!DOCTYPE>, html, head, body
  - Activity: Setting up a basic HTML document
- **Class - 02: Working with Text & Content**
  - Headings (<h1> to <h6>)
  - Paragraphs (<p>)
  - Lists (ordered <ol> & unordered <ul>)
  - Images (<img>) and links (<a>)
  - Assignment: Build a simple personal webpage with basic content.
- **Class - 03: Tables & Forms in HTML**
  - Creating tables: <table>, <tr>, <th>, <td>
  - Basic table formatting and accessibility
  - Creating forms: <form>, <input>, <label>, <button>
  - Different input types and form attributes
  - Activity: Build a simple contact form.

# HTML & CSS

## Week 02: Introduction to CSS Styling

- **Class - 04: CSS Fundamentals & Working with Backgrounds & Images**
  - What is CSS and why use it?
  - CSS Syntax: selectors, properties, and values
  - Linking CSS to HTML: internal, external, and inline styles
  - Basic styling: colors, fonts, text formatting
  - Applying background colors and images
  - Background properties: repeat, position, size, attachment
  - Using images effectively: optimization and responsiveness
  - Activity: Style the personal webpage created in Week 1, including background images.
- **Class - 05: The Box Model & Layout**
  - Understanding the CSS Box Model: content, padding, border, margin
  - Display property: block, inline, inline-block
  - Basic positioning: relative, absolute, fixed
  - Different input types and form attributes
  - Assignment: Create a two-column layout for a blog website.

# HTML & CSS

## • Class - 06: Grid and Flex Layout

- Introduction to CSS Grid: creating rows, columns, and areas.
- Understanding the CSS Grid layout model and its advantages.
- Introduction to CSS Flexbox: flex container and flex items.
- Building layouts with Flexbox.
- Activity: Recreate the two-column layout from Class 5 using Grid and Flexbox.

## Week 03: Responsive Design & Navigation

### • Class - 07: Introduction to Responsive Web Design

- What is responsive web design and why is it important?
- Mobile-first vs. desktop-first approaches
- Fluid grids and flexible images
- Activity: Analyze responsive and non-responsive websites.

### • Class - 08: CSS Media Queries

- Using media queries to target different screen sizes
- Defining breakpoints for responsive layouts
- Adapting layouts and content for different devices
- Assignment: Make the blog website responsive using media queries.

# HTML & CSS

- **Class - 09: Building Navigation Menus**

- HTML structure for navigation: <nav>, <ul>, <li>, <a>
- Styling navigation menus with CSS
- Creating dropdown menus and interactive elements
- Activity: Implement a responsive navigation menu for the blog website.

## Week 04: Expanding Your CSS Toolkit

- **Class - 10: Introduction to CSS Frameworks**

- What are CSS frameworks and their benefits?
- Overview of Bootstrap: grid system, components, utilities
- Getting started with Bootstrap: downloading and installation
- Activity: Build a basic webpage layout using Bootstrap.

- **Class - 11: Introduction to Tailwind CSS**

- What is Tailwind CSS and its utility-first approach?
- Setting up Tailwind CSS in a project
- Building layouts and styling elements with Tailwind's utility classes
- Activity: Recreate the Bootstrap webpage layout from Class 10 using Tailwind CSS.

- Class - 12: Other CSS Frameworks and Tools

- Introduction to CSS Variables
- Introduction to SASS
- Exploring other CSS frameworks: Bulma, Matcha.css, and their unique features.
- Resources and guidance for further exploration of CSS tools and frameworks.

## Git Essentials

By the end of this week, you will be able to:

- Understand the core concepts of version control and its importance in web development.
- Use basic Git commands to manage code changes, create branches, and revert to previous versions.
- Collaborate on projects using GitHub repositories, including forking, cloning, pushing, and pulling changes.
- Deploy static websites to hosting platforms like Github Pages, Netlify and Vercel.
- Understand the differences between various static site hosting options and their strengths.

# Git Essentials

## Week 05: Understanding Git, Github and Static Site Deployments

- Class - 01: Mastering Version Control with Git

- **Introduction to Version Control:**

- What is version control and why is it essential for developers?
    - Benefits of using a version control system like Git.
    - Understanding different version control systems (Git, SVN, Mercurial).

- **Getting Started with Git:**

- Installing Git on your local machine.
    - Configuring Git with your user information.
    - Basic Git commands: init, add, commit, status, log, diff.
    - Activity: Create a local Git repository, track changes to a file, and review commit history.

- **Branching and Merging in Git:**

- Understanding branches and their role in managing features and bug fixes.
    - Creating, switching, and deleting branches: branch, checkout, merge.
    - Merging branches and understanding merge conflicts.
    - Activity: Create a new branch, make changes, and merge it back to the main branch.

# Git Essentials

- **Undoing Changes and Reverting to Previous States:**
  - Understanding how to undo changes with git checkout and git revert.
  - Resetting your local branch to a specific commit.
  - Activity: Experiment with undoing changes and reverting to a previous commit.

## • Class - 02: Collaborative Coding with GitHub

- **Introduction to GitHub:**
  - What is GitHub and how it facilitates collaboration.
  - Creating a GitHub account and setting up your profile.
  - Understanding repositories, branches, forks, and pull requests.
- **Connecting Git to GitHub:**
  - Generating SSH keys for secure communication.
  - Adding your SSH key to your GitHub account.
  - Cloning a repository from GitHub to your local machine.
  - Pushing local changes to a remote GitHub repository.
  - Activity: Create a repository on GitHub, clone it locally, make changes, and push them back to GitHub.
- **Collaboration Workflow with GitHub:**
  - Forking a repository and understanding its purpose.
  - Creating pull requests to propose changes to the original repository.
  - Reviewing and merging pull requests.
  - Activity: Fork a classmate's repository, make changes, and create a pull request.

# Git Essentials

- **Exploring GitHub Features:**

- Using GitHub Issues for bug tracking and feature requests.
- Understanding GitHub Actions for automating workflows.
- Exploring GitHub Pages for simple website hosting.

- **Class - 03: Deploying Your Websites**

- **Introduction to Static Site Deployment:**

- What is a static website and its advantages.
  - Understanding the difference between static and dynamic websites.
  - Overview of popular static site hosting platforms (Netlify, Vercel, GitHub Pages, etc.).
  - Activity: Deploy a basic HTML and CSS website to Github Pages.

- **Deploying with Netlify:**

- Creating a Netlify account and connecting it to your GitHub repository.
  - Configuring Netlify settings (custom domain, build commands, environment variables).
  - Understanding Netlify's continuous deployment workflow.
  - Activity: Deploy a basic HTML and CSS website to Netlify.

# Git Essentials

- **Deploying with Vercel:**

- Creating a Vercel account and connecting it to your GitHub repository.
- Understanding Vercel's project detection and automatic configuration.
- Deploying serverless functions and APIs on Vercel (optional).
- Activity: Deploy a simple JavaScript project to Vercel.

- **Choosing the Right Deployment Platform:**

- Comparing and contrasting different static site hosting options.
- Factors to consider when choosing a platform (cost, features, ease of use).
- Resources and guidance for further exploration of static site deployment.

# JavaScript Essentials

Upon successful completion of this section,  
you will be able to:

- Understand the core concepts of JavaScript syntax, data types, and control flow.
- Write clean, efficient, and reusable JavaScript code.
- Work with objects, classes, and prototypes in JavaScript.
- Manipulate the DOM to create interactive web experiences.
- Make API requests to interact with external data and services.

## Week 06: JavaScript Fundamentals

### • Class - 01: Introduction to JavaScript

- What is JavaScript and what can you do with it?
- Setting up your JavaScript development environment.
- Writing your first JavaScript code: console logging, alerts.
- Basic syntax: variables, data types, operators, expressions.
- Activity: Create a simple JavaScript program that interacts with the user.

# JavaScript Essentials

- **Class - 02: Control Flow and Functions**
  - Conditional statements: if, else if, else, ternary operator.
  - Loops: for, while, do...while, for...in, for...of.
  - Functions: defining, invoking, parameters, return values.
  - Scope and closures.
  - Activity: Build a program that uses loops and conditional statements to solve a problem.
- **Class - 03: Working with Arrays and Objects**
  - Arrays: creating, accessing elements, iterating, common methods.
  - Objects: creating, properties, methods, iterating.
  - JSON: understanding and working with JSON data.
  - Activity: Create a program that manages data using arrays and objects.

## Week 07: Object-Oriented JavaScript and the DOM

- **Class - 04: Object-Oriented Programming in JavaScript**
  - Objects: review, constructor functions, this keyword.
  - Prototypes and inheritance.
  - ES6 Classes: syntax, inheritance, static methods.
  - Activity: Build a simple application using object-oriented principles.

# JavaScript Essentials

- **Class - 05: Object-Oriented Programming in JavaScript**
  - What is the DOM (Document Object Model)?
  - Selecting DOM elements: getElementById, querySelector.
  - Modifying DOM content and attributes.
  - Activity: Create a webpage and dynamically update its content using JavaScript.
- **Class - 06: DOM Events and Event Handling**
  - Event listeners: adding and removing listeners.
  - Event types: click, mouseover, submit, keypress.
  - Event object and event propagation.
  - Activity: Build an interactive webpage that responds to user events.

## Week 08: Asynchronous JavaScript and APIs

- **Class - 07: Advanced DOM Manipulation**
  - Creating and inserting DOM elements dynamically.
  - Traversing the DOM: parent, child, sibling nodes.
  - Working with forms: validation, submission handling.
  - Activity: Build a dynamic form with validation using JavaScript.

# JavaScript Essentials

- Class - 08: Asynchronous JavaScript: Callbacks, Promises, and Async/Await

- Understanding synchronous vs. asynchronous operations.
- Callbacks: passing functions as arguments.
- Promises: creating, resolving, rejecting, chaining.
- Async/Await: writing asynchronous code that looks synchronous.
- Error handling: try...catch blocks, throwing errors.
- Implementing error handling with Promises and Async/Await.
- Activities:
  - Make an asynchronous request using promises and then refactor it using async/await.
  - Write a program that handles errors gracefully during asynchronous operations.

- Class - 09: Making API Requests and Handling Responses

- What is an API (Application Programming Interface)?
- Making GET requests using the Fetch API.
- Making POST requests to send data to an API.
- Handling API responses: status codes, JSON parsing.
- Activity: Fetch data from a public API and display it on a webpage. Then, create a new resource on the API using a POST request.

# JavaScript Essentials

## Week 09: API Authentication, Testing & Final Project

- **Class - 10: API Authentication and Best Practices**

- API authentication: API keys, basic authentication, OAuth.
- Handling API rate limiting and errors effectively.
- Best practices for working with APIs in JavaScript applications.
- Activity: Interact with an API that requires authentication.

- **Class - 11: Introduction to JavaScript Unit Testing**

- What is unit testing and why is it important?
- Understanding test-driven development (TDD).
- Setting up a testing environment: Jest framework introduction.
- Writing basic unit tests for JavaScript functions.
- Activity: Write unit tests for a simple JavaScript function.

- **Class - 12: Developer Console**

- Introduction to browser developer tools (Chrome DevTools).
- Debugging JavaScript code: setting breakpoints, inspecting variables.
- Analyzing network requests: inspecting headers, responses, and performance.
- Using the Elements panel to debug HTML and CSS.
- Activity: Debug a React application using Chrome DevTools.

# React Essentials

+ Upon successful completion of this section,  
you will be able to:

- Understand the core concepts of React, including components, props, and state.
- Build interactive and dynamic user interfaces using React components.
- Manage application state efficiently using React's built-in mechanisms and external libraries like Redux and Zustand.
- Fetch data from APIs and integrate it seamlessly into your React applications.
- Write unit and integration tests for your React components to ensure code quality.
- Build a complete application using React.

# React Essentials

## Week 10: Introduction to React Fundamentals

- **Class - 01: Getting Started with React**
  - What is React and why is it so popular?
  - Setting up your React development environment (Create React App).
  - Understanding JSX syntax and its role in React.
  - Building your first React component: Hello World.
  - Activity: Create a simple React application with multiple components.
- **Class - 02: Components, Props, and State**
  - Deep dive into React components: functional vs. class components.
  - Passing data between components using props.
  - Managing component state with useState hook.
  - Understanding component lifecycle methods (for class components).
  - Activity: Build a counter application with multiple components and state management.

# React Essentials

- **Class - 03: Working with Lists, Conditional Rendering, and useEffect**
  - Rendering lists of data in React using map().
  - Implementing conditional rendering based on props and state.
  - Using array keys for efficient list rendering.
  - Introduction to the useEffect hook:
    - Performing side effects in functional components (fetching data, DOM manipulation).
    - Understanding the dependency array and its role in controlling side effects.
  - Activity: Build an application with adding, deleting, marking items as complete, and incorporating use Effect for local storage persistence.

## Week 10: Introduction to React Fundamentals

- **Class - 04: Event Handling and Forms in React**
  - Handling user events (clicks, input changes) in React.
  - Creating controlled forms with state management.
  - Implementing form validation.
  - Activity: Build a registration form with input validation.

# React Essentials

- **Class - 05: Styling React Components**

- Inline styles vs. CSS Modules vs. styled-components.
- Applying CSS Modules for modular and maintainable styling.
- Introduction to styled-components for CSS-in-JS approach.
- Activity: Style a React application using CSS Modules or styled-components.

- **Class - 06: React Router and Authenticated Routes**

- Introduction to single-page applications (SPAs).
- Setting up client-side routing with React Router.
- Creating routes, links, and navigating between different views.
- Implementing protected routes and authentication:
  - Using conditional rendering or route guards to restrict access to certain routes.
  - Introduction to basic authentication concepts.
- Activity: Build a multi-page application with React Router, including a home page, about page, and a protected dashboard accessible only after simulated login.

# React Essentials

## Week 12: Working with APIs, Advanced Hooks, and State Management

- Class - 05: Fetching Data from APIs
  - Making API requests in React using fetch or Axios.
  - Handling asynchronous data fetching with useEffect hook.
  - Displaying loading states and handling errors.
  - Activity: Fetch data from a public API and display it in a React component.
- Class - 08: Advanced Hooks and Context API
  - Deep dive into advanced hooks:
    - useReducer: Managing complex state logic with a reducer function.
    - useContext: Accessing shared state from Context API.
    - useRef: Creating references to DOM elements and storing values.
    - useCallback: Optimizing performance by memoizing callback functions.
    - useMemo: Memoizing expensive calculations to prevent redundant computations.

# React Essentials

- **Implementing Context API for global state management:**
  - Creating context providers and consumers.
  - Passing data through the component tree without prop drilling.
- **Activity: Build a feature using useReducer for state management and Context API to share state between components.**
- **Class - 09: Introduction to External State Management: Redux and Zustand**
  - Understanding the need for centralized state management beyond simple props and Context API.
  - **Introduction to Redux:**
    - Actions, reducers, store, and connecting components.
    - Implementing asynchronous actions with Redux Thunk or Redux Saga.
  - **Exploring Zustand as a lightweight alternative for state management:**
    - Setting up the store, defining actions, and using the store within components.
    - **Activity: Refactor a previous application feature to use either Redux or Zustand for state management.**

# React Essentials

## Week 12: Working with APIs, Advanced Hooks, and State Management

- Class - 10: Testing and Final Project
  - Introduction to testing React components with Jest and React Testing Library.
  - Writing unit tests for component logic and rendering.
  - Simulating user interactions and testing event handlers.
  - Testing asynchronous code and API interactions.
  - Activity: Write unit tests for a React component, including tests for asynchronous operations and custom hooks.
- Class - 11 & 12: Final Project - Building a React Application
  - Project Planning: Brainstorming project ideas and defining scope.
  - Project Development: Implementing the chosen project, applying learned concepts, and choosing a state management solution based on project needs.
  - Code Review and Feedback: Presenting projects, receiving feedback, and iterating on implementations.

# Next JS Essentials

## Prerequisites:

- Solid understanding of React fundamentals (components, props, state, hooks).
- Familiarity with JavaScript ES6+ syntax.
- Basic knowledge of HTML, CSS, and web development concepts.

## Upon successful completion of this course, you will be able to:

- Understand the core concepts and benefits of Next.js for building modern web applications.
- Build server-side rendered React applications with improved SEO and performance.
- Implement various data fetching techniques in Next.js for dynamic content.
- Create API routes with Next.js for serverless functions.
- Utilize Next.js features like routing, image optimization, and pre-rendering.
- Deploy Next.js applications to production environments.

# Next JS Essentials

## Week 14: Introduction to Next.js and Server-Side Rendering

- **Class - 01: Getting Started with Next.js**
  - Why Next.js? Benefits over traditional React applications.
  - Setting up a Next.js project.
  - Understanding the file-based routing system.
  - Creating and rendering basic components.
  - Activity: Build a simple multi-page website with Next.js routing.
- **Class - 02: Introduction to Server-Side Rendering (SSR)**
  - What is SSR and its advantages for SEO and performance?
  - Implementing SSR with `getServerSideProps` in Next.js.
  - Fetching data on the server and passing it to components.
  - Understanding the differences between client-side and server-side rendering.
  - Activity: Convert a static page to use SSR and fetch data from an external API.

# Next JS Essentials

- Class - 03: Data Fetching Techniques in Next.js
  - Exploring different data fetching methods:
    - getStaticProps: Fetching data at build time.
    - getServerSideProps: Fetching data on each request.
    - SWR and React Query: Client-side data fetching and caching.
  - Choosing the right data fetching strategy based on application requirements.
  - Activity: Implement a blog page using different data fetching techniques for various sections.

## Week 15: API Routes, Dynamic Routing, and Advanced Features

- Class - 04: Building API Routes with Next.js
  - Introduction to API routes and serverless functions in Next.js.
  - Creating API endpoints to handle data requests.
  - Connecting to databases and external APIs within API routes.
  - Activity: Build an API route to handle user registration and store data in a database.

# Next JS Essentials

- **Class - 05: Dynamic Routing and Data Fetching**
  - Implementing dynamic routes with route parameters in Next.js.
  - Fetching data based on dynamic route segments.
  - Handling loading states and error scenarios.
  - Activity: Create a product details page with dynamic routing and data fetching based on the product ID.
- **Class - 06: Advanced Next.js Features**
  - Image optimization with next/image.
  - Pre-rendering strategies for static site generation (SSG) and incremental static regeneration (ISR).
  - Optimizing Next.js applications for performance and SEO.
  - Activity: Implement image optimization and explore SSG or ISR for a blog or e-commerce project.

# Next JS Essentials

## Week 16: Authentication, Deployment, and Final Project

- **Class - 07: Authentication in Next.js**
  - Implementing user authentication with Next.js API routes.
  - Protecting routes and handling user sessions.
  - Integrating with authentication providers (e.g., Auth0, Firebase Authentication).
  - Activity: Add authentication to an existing Next.js application.
- **Class - 08 & 09: Final Project - Building a full-stack application with Next.js**
  - Project Planning: Brainstorming project ideas and defining scope.
  - Project Development: Implementing the chosen project, applying learned concepts, and choosing appropriate technologies and libraries.
  - Deployment and Presentation: Deploying the project to a live environment and presenting the final product.

# Mongo DB Essentials

## Prerequisites:

- Basic understanding of JavaScript and programming concepts.
- Familiarity with databases and data manipulation concepts is beneficial.

## Upon successful completion of this course, you will be able to:

- Understand the fundamentals of NoSQL databases and MongoDB's document-based model.
- Design and implement effective schemas using Mongoose for various data structures.
- Perform CRUD (Create, Read, Update, Delete) operations on MongoDB data using Mongoose methods.
- Utilize advanced querying techniques, including filtering, sorting, and aggregation.
- Implement data validation and relationships between documents.
- Understand and apply best practices for performance optimization and data integrity.

# Mongo DB Essentials

## Week 17: MongoDB Fundamentals and Mongoose Basics

- **Class - 01: Introduction to NoSQL and MongoDB**
  - Understanding NoSQL databases and their use cases.
  - Exploring MongoDB's document-based data model and its advantages.
  - Installing and setting up MongoDB locally.
  - Basic MongoDB shell commands for interacting with data.
  - Activity: Create a sample database and collections using MongoDB shell commands.
- **Class - 02: Introduction to Mongoose and Schema Design**
  - What is Mongoose and its role in working with MongoDB?
  - Setting up a Node.js project with Mongoose.
  - Defining schemas and models for structuring data.
  - Understanding different data types and validation options in Mongoose.
  - Activity: Design a Mongoose schema for a blog post, including title, content, author, tags, and timestamps.

# Mongo DB Essentials

- Class - 03: CRUD Operations with Mongoose
  - Creating, reading, updating, and deleting documents using Mongoose methods.
  - Handling asynchronous operations with promises and async/await.
  - Implementing data validation and error handling.
  - Activity: Build a simple command-line application to perform CRUD operations on your blog post model.

## Week 18: Advanced Queries, Relationships, and Best Practices

- Class - 04: Advanced Querying Techniques with Mongoose
  - Filtering data with comparison, logical, and element operators.
  - Sorting and limiting query results.
  - Using projection to retrieve specific fields.
  - Understanding and utilizing text indexes for search functionality.
  - Activity: Implement various queries to filter and retrieve blog posts based on different criteria.

# Mongo DB Essentials

- **Class - 05: Relationships and Embedded Documents in Mongoose**
  - Defining relationships between documents using references and embedding.
  - Understanding the trade-offs between referencing and embedding.
  - Implementing one-to-one, one-to-many, and many-to-many relationships.
  - Activity: Add comments functionality to your blog, implementing a one-to-many relationship between posts and comments.
- **Class - 06: Aggregation, Performance, and Best Practices**
  - Introduction to aggregation pipelines for data processing and analysis.
  - Performing calculations, grouping data, and reshaping results with aggregation.
  - Understanding indexes and their importance for query performance.
  - Best practices for schema design, data modeling, and query optimization.
  - Activity: Utilize aggregation to generate reports or insights from your blog data.

# MySQL Essentials

## Prerequisites:

- Basic understanding of databases and data management concepts.
- Familiarity with basic programming concepts is beneficial.

## Upon successful completion of this section, you will be able to:

- Confidently design and implement relational databases using MySQL.
- Write efficient SQL queries to retrieve, manipulate, and analyze data.
- Understand and apply database normalization principles.
- Work with indexes and transactions for optimized performance.
- Utilize ORMs like Sequelize or Prisma to interact with MySQL databases from your applications.

# MySQL Essentials

## Week 19: MySQL Foundations: Design, Queries, and Data Manipulation

- **Class - 01: Diving into the World of Databases**
  - What are databases, and why are they essential?
  - Relational vs. NoSQL databases: understanding the differences.
  - Introduction to MySQL: key features, benefits, and common use cases.
  - Activity: Install MySQL Server and a database client (e.g., MySQL Workbench, DBeaver).
- **Class - 02: Blueprinting Your Data: Database Design Fundamentals**
  - Understanding entities, attributes, and relationships in database design.
  - Creating tables: defining data types, primary keys, and constraints for data integrity.
  - Visualizing relationships: Introduction to Entity-Relationship Diagrams (ERDs).
  - Activity: Design a simple database schema for a real-world scenario like a blog or e-commerce store.

# MySQL Essentials

- **Class - 03: Speaking the Language of Data: SQL Fundamentals**
  - Introduction to SQL (Structured Query Language): the backbone of relational databases.
  - Mastering data retrieval: SELECT, FROM, WHERE clauses for precise data selection.
  - Refining your queries: Filtering and sorting data using operators, ORDER BY, and LIMIT.
  - Activity: Write SQL queries to extract specific information from your designed database

## Week 20: Advanced SQL, Data Integrity, and ORMs

- **Class - 04: Data Manipulation: CRUD Operations with SQL**
  - Populating your database: INSERT INTO statement for adding new records.
  - Modifying existing data: UPDATE statement for making changes to records.
  - Removing data: DELETE statement for deleting unwanted records.
  - Activity: Populate your database with sample data and practice data manipulation techniques.

# MySQL Essentials

- **Class - 05: Guardians of Data: Constraints and Data Integrity**

- Enforcing data rules: NOT NULL, UNIQUE, CHECK, DEFAULT values for maintaining data consistency.
- Relationships matter: Foreign keys and referential integrity for linking related data.
- Data validation techniques: Ensuring data accuracy and reliability.
- Activity: Implement various constraints and data validation rules in your database schema.

- **Class - 06: Unifying Data: Joins and Subqueries**

- Combining data from multiple tables: INNER JOIN, LEFT JOIN, RIGHT JOIN for powerful data retrieval.
- Nesting queries: Subqueries (nested SELECT statements) for more complex data filtering.
- Activity: Write SQL queries involving joins and subqueries to answer intricate questions from your database.

# MySQL Essentials

## Week 21: Performance Optimization, ORMs, and Project Development

- **Class - 07: Boosting Performance: Indexing and Transactions**
  - Speeding up your queries: Understanding indexes, their types, and their impact on query performance.
  - Creating and managing indexes: Applying indexing strategies for optimal data retrieval.
  - Ensuring data consistency: ACID properties of transactions and their importance.
  - Managing transactions: START TRANSACTION, COMMIT, ROLLBACK for reliable data operations.
  - Activity: Experiment with indexing to optimize queries and implement transactions to maintain data integrity.
- **Class - 08: ORMs in Action: Building Applications with MySQL and [Sequelize/Prisma]**
  - Deep dive into your chosen ORM: Defining models, relationships, and performing database operations.
  - Building a simple application: Integrating your MySQL database with a basic application using the ORM.
  - Activity: Create a small application feature that interacts with your database using the ORM.

# MySQL Essentials

- Class - 09: Final Project: Building a Database-Driven Application
  - Project Planning: Choose a project idea that involves building an application with a MySQL database (e.g., a to-do list app, or a movie app, or a library management system).
  - Project Development: Design and implement your database, write SQL queries or utilize the ORM, and build the application logic.

# NodeJS and Express

## Prerequisites:

- Strong JavaScript Foundation: Deep understanding of ES6+ features, asynchronous programming, and common patterns.
- Database Proficiency: Experience with MongoDB, Mongoose, and/or MySQL is beneficial but not required.
- Testing Mindset: Familiarity with unit testing principles is helpful but not mandatory.

## By the end of this section, you will be able to:

- Confidently build robust and scalable web applications and APIs using Node.js and Express.js.
- Understand and apply essential security principles to protect your Node.js applications.
- Develop, test, document, and publish production-ready Node.js packages.
- Join the thriving Node.js community and confidently contribute your own packages.

# NodeJS and Express

## Week 22: Introduction to Node.js and Building Web Servers with Express.js

### • Class - 01: Unleashing the Power of Node.js

- What is Node.js and why is it so popular?
- Understanding the Node.js architecture and the event loop.
- Setting up your development environment: installing Node.js and npm.
- Modules, packages, and npm: organizing your code and using third-party libraries.
- Activity: Create a simple Node.js script that interacts with the file system.

### • Class - 02: Asynchronous Programming in Node.js

- Deep dive into callbacks, Promises, and async/await.
- Error handling and propagation in asynchronous code.
- Activity: Build a Node.js script that fetches data from multiple APIs concurrently and handles errors gracefully.

### • Class - 03: Building Your First Express.js Application

- What is Express.js and why use a web framework?
- Setting up an Express.js project and understanding the basic structure.
- Routing, middleware, and handling different HTTP methods (GET, POST, PUT, DELETE).
- Activity: Create a simple "Hello World" application and practice basic routing with Express.js.

# NodeJS and Express

## Week 23: Express.js in Action: Building a RESTful API

- **Class - 04: Designing and Building a RESTful API**
  - RESTful API design principles: resources, HTTP methods, status codes.
  - Building a RESTful API with Express.js to manage a collection of data (e.g., products, blog posts).
  - Activity: Create a complete RESTful API for a resource of your choice (e.g., a to-do list, a book library).
- **Class - 05: Working with Databases (MongoDB and Mongoose)**
  - Introduction to NoSQL databases and MongoDB.
  - Using Mongoose for object-document mapping and schema validation.
  - Integrating MongoDB and Mongoose with your Express.js API.
  - Activity: Connect your API from the previous class to a MongoDB database to persist data.
- **Class - 06: Authentication and Authorization in Express.js**
  - Implementing user authentication with Passport.js.
  - Strategies for local authentication and JWTs (JSON Web Tokens).
  - Protecting API routes based on user authentication.
  - Activity: Add basic user authentication to your Express.js API using JWTs.

# NodeJS and Express

## Week 24: Securing Your API and Handling Errors

- **Class - 07: API Security Best Practices**
  - Input validation and sanitization: preventing common security vulnerabilities.
  - Rate limiting and protecting against brute-force attacks.
  - Securely storing and managing sensitive data (passwords, API keys).
  - Activity: Perform a security audit and implement best practices on your existing API.
- **Class - 08: Error Handling and Debugging in Node.js**
  - Best practices for error handling in asynchronous code.
  - Using the built-in debugger and other debugging tools.
  - Logging and monitoring your Node.js applications.
  - Activity: Implement robust error handling and logging in your Express.js API.
- **Class - 09: Project Time: Building a Real-World API**
  - Project Planning: Choose a project idea that involves building a practical API (e.g., a simple e-commerce API, a social media API).
  - Project Development: Build your API, focusing on code quality, API design, and security best practices.

# NodeJS and Express

## Week 25: Node.js Package Development: Sharing Your Code

- **Class - 10: Entering the World of Node.js Packages**
  - Understanding the Node.js package ecosystem and npm.
  - Exploring popular Node.js packages and their uses.
  - Identifying opportunities to create your own packages.
  - Activity: Research and present on a Node.js package that you find interesting.
- **Class - 11: Creating Your First Node.js Package**
  - Structuring your package: modules, dependencies, and entry points.
  - Writing clean, reusable, and well-documented code.
  - Publishing your package to the npm registry.
  - Activity: Develop and publish a simple Node.js package that provides a utility function.
- **Class - 12: Testing and Maintaining Your Package**
  - Writing unit tests for your package using Jest or another testing framework.
  - Semantic versioning, handling updates, and managing dependencies.
  - Activity: Write unit tests for your package and explore advanced publishing options.

# NodeJS and Express

## Week 26: Final Project and Course Wrap-Up

- Class - 01: Final Project: Building Your Own Package
  - Project Time!
    - Project Development: Continue working on your Node.js package, focusing on code quality, reusability, and documentation. Ensure it is ready for presentation and potentially publishing to the npm registry.
- Class - 02: Node.js & Express Mastery Recap
  - Core Concepts Revisited: Review key Node.js concepts, including the event loop, modules, asynchronous programming, and error handling.
  - Best Practices and Common Pitfalls: Discuss best practices for building Node.js applications and common pitfalls to avoid.
  - Q&A Session: Address any remaining questions about Node.js and its ecosystem.

# NodeJS and Express

- **Class - 03: Your Node.js Journey and Beyond**

- Course Summary and Next Steps: Recap the key takeaways from the course and provide guidance on further learning resources and paths.
- Career Paths MERN Stack: Explore different career opportunities available for MERN Stack developers, such as back-end developer, full-stack developer, or DevOps engineer.
- Building Your Portfolio and Networking: Discuss strategies for showcasing your Node.js skills, building a portfolio, and networking within the Node.js community.

যেকোনো সমস্যায় পড়লে

# কল করুন

আমাদের কাস্টমার কেয়ার নাম্বারে:



+880 9613-829043



Scan to visit our website

